Channel Assignment in Cellular Radio

Kumar N. Sivarajan and Robert J. McEliece,
*California Institute of Technology,**
and
John W. Ketchum,
GTE Laboratories Incorporated.

Abstract

In this paper, we describe some heuristic channel assignment algorithms for cellular systems, that we have recently developed. These algorithms have yielded optimal, or near-optimal assignments, in many cases. The channel assignment problem can be viewed as a generalized graph coloring problem, and these algorithms have been developed, in part, by suitably adapting some of the ideas previously introduced in heuristic graph coloring algorithms.

Introduction

With the growth in demand for mobile telephone services and the limited allocation of spectrum for this purpose, the problem of optimal assignment of frequency channels, in order to make the most efficient use of the available spectrum, is becoming increasingly important. In this paper we describe channel assignment algorithms which we have recently developed and which have performed well in many of the examples we considered. We formulate the channel assignment problem as a minimum span problem, i.e., a problem wherein we are required to find the minimum bandwidth necessary to satisfy a given demand.

Problem Statement

Frequencies are represented by the positive integers 1, 2, 3, ...

Given :

N : the number of cells in the system

m_i, $1 \leq i \leq N$: the number of channels required in cell i

c_{ij}, $1 \leq i,j \leq N$: the frequency separation required between a call in cell i and a call in cell j

Find :

f_{ik}, $1 \leq i \leq N$, $1 \leq k \leq m_i$: the frequency assigned to the kth call in the ith cell

such that,

$$\max_{i,k} f_{ik}$$

(i.e. the total number of frequencies required), is a minimum, subject to the separation constraints,

$$|f_{ik} - f_{jl}| \geq c_{ij}$$

for all i, j, k, l except for $i = j$, $k = l$.

Example 1. The number of cells is $N = 4$. $\mathbf{m} = (m_i) = (1, 1, 1, 3)$ is the vector of requirements. The separation matrix $\mathbf{C} = (c_{ij})$ is

$$\mathbf{C} = \begin{pmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{pmatrix}.$$

It is required to find positive integers (frequencies) f_{11}, f_{21}, f_{31}, f_{41}, f_{42} and f_{43}, such that their maximum is a minimum, subject to the separation constraints specified by $\mathbf{C}$.

This problem is equivalent to the following generalized graph coloring problem. Consider the graph obtained by representing each call by a vertex, with an edge joining two vertices if the corresponding calls cannot use the same frequency. This edge is labelled with the required mimimum separation between the frequencies assigned to these calls. The frequency assignment problem is then equivalent to assigning positive integers to the vertices of this graph such that, if two vertices are connected by an edge, the absolute value of the difference of the integers assigned to these vertices, is at least equal to the edge label, and, the maximum integer used is as small as possible. If all the c_{ij}'s are 0's and 1's (pure co-channel case), this reduces to the classical graph coloring problem. Since the latter is known to be NP-complete, it follows that the generalized graph coloring problem is also NP-complete [6].

The basic idea of all of our algorithms is to list the calls in some order, and use either a requirement exhaustive strategy or a frequency exhaustive strategy. (See [2] or [3]).

*The contribution of K. N. Sivarajan and R. J. McEliece to this paper was supported by a grant from GTE Laboratories.

CH2379-1/89/0000/0846 $1.00 © 1989 IEEE

Frequency Exhaustive strategy

1. Starting at the top of the list, assign to each call the *least* possible frequency, consistent with previous assignments i.e., without violating the separation constraints.

Requirement Exhaustive strategy

1. Take frequency 1 and assign it to the first call in the list. There may be other calls, further down the list, which can reuse frequency 1. If so, assign frequency 1 again to the first such call in the list. Continue in this manner until there is no call in the list, to which frequency 1 can be assigned.
2. Now take frequency 2, and starting at the top of the list, similarly assign it to all possible calls in the list.
3. Continue in this manner until all the calls have been assigned frequencies. (In this strategy, one takes a frequency and exhausts the requirements (calls); hence the name.)

The *degree* of cell i is defined as

$$d_i = \left(\sum_{j=1}^{N} m_i c_{ij}\right) - c_{ii}, \qquad 1 \leq i \leq N,$$

which is a heuristic measure of the difficulty of assigning a frequency to a call in that cell. The *degree of a call* is the degree of the cell in which it is contained. In the equivalent graph coloring problem described above, the degree of a call is equal to the sum of the labels on the edges, incident at the vertex corresponding to the call.

Based on this, two different orderings of the cells are considered. They are the Node-color and Node-degree orderings considered by Zoellner and Beall in [2], except that the *above* definition of the *degree* of a cell is used. In the Node-degree ordering, the cells are arranged in decreasing order of their degrees. The Node-color ordering is obtained as follows : Of the N cells, the cell with the least degree is placed at the last (Nth) place in the list. This cell is eliminated from the system and the degrees of the remaining cells are recomputed. Now, the cell with the least degree is placed at the $(N-1)$th position in the list, and eliminated from the system. This process is continued until the ordering is complete. These orderings are modifications of the 'highest degree first' and 'least degree last' heuristics in graph coloring.

Once the *cells* have been ordered, the *calls* can be ordered in two ways. The calls are arranged in an $(N \times m_{\max})$ matrix, where N is the number of cells and $m_{\max}$ is the maximum number of calls in any cell. Each row of the matrix corresponds to the calls in a cell. The rows are arranged in Node-color or Node-degree order as explained above. The idea is to arrange the calls such that all the columns have nearly the same number of calls. Calls in the first row start at the first column. Calls in the second row start at column (m_1+1), if the first row has m_1 calls, and cyclically fill this row. Similarly, calls in the third row start where the second ends and so on.

Example 1 (continued). The degrees of the calls are $\mathbf{d} = (4, 7, 6, 13)$. Therefore the Node-degree ordering is (cell 4, cell 2, cell 3, cell 1). The matrix of calls corresponding to this is

$$\mathbf{A}_d = \begin{pmatrix} a_{41} & a_{42} & a_{43} \\ a_{21} & & \\ & a_{31} & \\ & & a_{11} \end{pmatrix}.$$

In the Node-color ordering, cell number 1 is again the last in the list since it has the least degree. If this cell is eliminated, the degrees of the other cells become $\mathbf{d} = (-, 3, 6, 13)$. Therefore, cell number 2 will be in the third place in the list. The final Node-color ordering is (cell 4, cell 3, cell 2, cell 1). The matrix of calls corresponding to this is

$$\mathbf{A}_c = \begin{pmatrix} a_{41} & a_{42} & a_{43} \\ a_{31} & & \\ & a_{21} & \\ & & a_{11} \end{pmatrix}.$$

Once the calls have been so arranged in a matrix, two orderings of the calls are obtained by either listing all the calls in the first row, then the second, and so on (Row-wise ordering), or listing the calls in the first column, then the second, and so on (Column-wise ordering). Therefore one obtains *four* ways of ordering the *calls* from *two* ways of ordering the *cells*. Combined with *two* techniques of assigning frequencies, this gives rise to *eight* frequency assignment algorithms.

The assignments obtained using one or the other of these algorithms, on many of the examples we considered, is close to the best lower bound (LB) obtained using the lower bounds in [4].

Example 1 (again). Consider the matrix of calls $\mathbf{A}_d$. Ordering the calls row-wise, one obtains (a_{41}, a_{42}, a_{43}, a_{21}, a_{31}, a_{11}) as the list of calls. A frequency exhaustive strategy applied to this list of calls gives the frequency assignment (1, 6, 11, 2, 3, 6). A requirement exhaustive strategy, at the first step, assigns frequency 1 to calls 1 and 7 (a_{41} and a_{11}). The complete assignment using this strategy is (1, 6, 11, 5, 3, 1). The maximum frequency used by both these assignments is 11, which is also the lower bound. This is because, any two of the three calls in cell number 4 require a separation of 5 between them. Ordering the calls column-wise, one gets (a_{41}, a_{21}, a_{42}, a_{31}, a_{43}, a_{11}). The row-wise and column-wise ordering methods applied to the matrix of calls $\mathbf{A}_c$ yield (a_{41}, a_{42}, a_{43}, a_{31}, a_{21}, a_{11}) and (a_{41}, a_{31}, a_{42}, a_{21}, a_{43}, a_{11}), respectively. In all the above cases, in this particular example, a frequency exhaustive strategy gives the assignment (a_{11} : 6, a_{21} : 2, a_{31} : 3, a_{41} : 1, a_{42} : 6, a_{43} : 11) whereas a requirement exhaustive strategy gives the assignment (a_{11} : 1, a_{21} : 5, a_{31} : 3, a_{41} : 1, a_{42} : 6, a_{43} : 11).

It is important to note that all the above algorithms are non-iterative, and hence fast ($O(n^2)$ where n is the total number of channel requirements), compared to iterative algorithms like the one proposed by Box [1]. This feature is particularly important in the case of large cellular systems. This, and the fact that these algorithms are applicable to any cellular system (not necessarily consisting of regular, hexagonal cells), are important when one is trying to choose the optimal locations for the cell sites, by repeated application of a channel assignment algorithm, since a large number of cases may have to be solved.

Algorithm Performance Results

The performance of these algorithms is shown in Tables 2 and 3 for various sets of constraints. The cellular system considered is the 21-cell example found in [4], which is reproduced as Figure 1. In this figure, the cell number is indicated within each cell. Table 1 (Case 1) reproduces the inhomogeneous requirements for this system given in [4]. These requirements were used to compile Table 2. Table 1 (Case 2) gives another set of channel requirements which were used in compiling Table 3.

In Tables 2 and 3, the best lower bound obtained using the bounds in [4] (LB), and the performance of our channel assignment algorithms are tabulated. The entries in the table are the number of frequencies (span) required. A three letter code is used to indicate the algorithms. The first letter is 'C' or 'D' and denotes 'Node-color order' or 'Node-degree order' respectively. The second letter is 'R' or 'C' for 'Row-wise' or 'Column-wise' ordering. The last letter is 'R' or 'F' for 'Requirement' or 'Frequency' Exhaustive method of assignment. A 2 (resp. 1) in column 'acc' implies the presence (resp. absence) of adjacent channel constraints on adjacent cells. The co-site constraint is indicated in column 'c_{ii}'. (The number closest to the lower bound in each row is in italics.)

Other Possible Orderings

If all the non-zero entries in the separation matrix are taken to be unity, the cells ordered using the Node-color or Node-degree ordering described above, and the calls ordered row-wise, we obtain the Node-color and Node-degree orderings described in [3]. The best assignment obtained using our algorithms, in all the examples considered, is better then the assignments obtained by using this ordering of the calls, and a frequency or requirement exhaustive strategy. However, the performance of the frequency assignment algorithms obtained by using this ordering of the cells, and column-wise ordering of calls, is better in some cases. One such case is in Table 2, $N_c = 7$, acc $= 2$ and $c_{ii} = 5$. The minimum number of frequencies used by any of the algorithms listed there is 447 but with the Node-color ordering of cells described in [3], *column-wise* ordering of calls and a frequency exhaustive assignment strategy, an assignment which uses only 445 frequencies may be obtained.

Therefore, the two new ideas on channel assignment introduced in this paper viz., the new definition of the degree of a cell (or call) in the presence of arbitrary constraints (not purely co-channel), and the column-wise ordering of calls, which corresponds essentially to taking a call from each cell in the system in succession (with some modification to accommodate the unequal numbers of calls in each cell), achieve significant savings in the spectrum needed for a frequency assignment problem.

One of the weaknesses of these algorithms that we have have noticed is their performance in the case of homogeneous requirements. To remedy this, we introduce yet another ordering of the cells which we call the Co-channel Sets ordering. This ordering is applicable only to regular, hexagonal cellular systems.

Co-channel Sets ordering

1. Assume the requirements are homogeneous and a regular frequency reuse plan is to be adopted. This necessitates dividing the available spectrum, and the cells in the system, into N_c classes, where N_c is an integer of the form $i^2 + ij + j^2$ and, i and j are non-negative integers. The most commonly used values of N_c are 4, 7 and 12.
2. Order the cells so that all the cells belonging to class 1 are at the top of the list, followed by those belonging to class 2, and so on.

If the requirements are homogeneous, with only co-channel and co-site constraints, column-wise ordering of calls, and a frequency exhaustive assignment strategy, will give an optimal assignment. In the presence of adjacent channel constraints, the numbering of the classes is important, but not otherwise. (See [5] for a complete treatment of the homogeneous frequency assignment problem in regular, hexagonal cellular systems). This ordering of the cells can be used to get good results in the case of near-homogeneous requirements. It is interesting to note that if the calls are ordered such that all calls which are assigned frequency 1, in an optimal assignment, are at the top of the list, followed by calls which are assigned frequency 2, and so on, either a frequency exhaustive or a requirement exhaustive strategy will come up with an optimal assignment (i.e, using the minimum number (span) of frequencies).

In addition to being NP-complete, graph coloring is one of the most difficult problems to develop approximation algorithms for. It is shown in [7] that the problem of finding a fast (polynomial-time) algorithm that guarantees a coloring using less than twice the minimum number of colors, is itself NP-complete. In the light of these results, the performance of the heuristics we have developed seems very good indeed.

References

[1] Box, F., "A Heuristic Technique for Assigning Frequencies to Mobile Radio Nets", *IEEE Trans. Veh. Tech.*, vol. VT-27, pp. 57-64, May 1977.

[2] Zoellner, J. A., and Beall, C. A., "A Breakthrough in Spectrum Conserving Frequency Assignment Technology", *IEEE Trans. Electromagn. Comp.*, vol. EMC-19, pp. 313-319, Aug. 1977.

[3] Gamst, A., and Rave, W., "On Frequency Assignment in Mobile Automatic Telephone Systems", *Proc. GLOBECOM '82*, pp. 309-315.

[4] Gamst, A., "Some Lower Bounds for a Class of Frequency Assignment Problems", *IEEE Trans. Veh. Tech.*, vol. VT-35, pp. 8-14, Feb. 1986.

[5] Gamst, A., "Homogeneous Distribution of Frequencies in a Regular, Hexagonal Cell System", *IEEE Trans. Veh. Tech.*, vol. VT-31, pp. 132-144, 1982.

[6] Garey, M. R., and Johnson, D. S., *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York, 1979.

[7] Garey, M. R., and Johnson, D. S., "The Complexity of Near-Optimal Graph Coloring", *Journal of the ACM*, vol. 23, pp. 43-49, Jan. 1976.

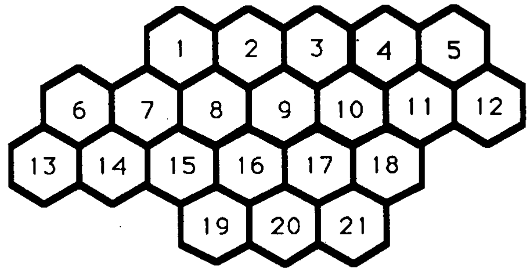

Figure 1: The 21-cell system (The cell number is indicated within each cell.)

Case ⟶	1	2
i	m_i	m_i
1	8	5
2	25	5
3	8	5
4	8	8
5	8	12
6	15	25
7	18	30
8	52	25
9	77	30
10	28	40
11	13	40
12	15	45
13	31	20
14	15	30
15	36	25
16	57	15
17	28	15
18	8	30
19	10	20
20	13	20
21	8	25

Table 1: The Channel Requirements in the Examples considered in Tables 2 and 3

N_c	acc	c_{ii}	LB	CRF	CRR	CCF	CCR	DRF	DRR	DCF	DCR
12	2	5	414	543	464	*460*	476	543	521	475	504
7	2	5	414	543	468	451	501	543	466	*447*	495
12	2	7	533	*536*	565	546	562	*536*	566	546	565
7	2	7	533	536	564	546	559	536	561	*533*	566
12	1	5	381	*381*	*381*	*381*	*381*	*381*	*381*	*381*	*381*
7	1	5	381	*381*	*381*	*381*	*381*	*381*	*381*	*381*	*381*
12	1	7	533	*533*	*533*	*533*	*533*	*533*	*533*	*533*	*533*
7	1	7	533	*533*	*533*	*533*	*533*	*533*	*533*	*533*	*533*

Table 2: Algorithm results for Case 1 requirements in Table 1

N_c	acc	c_{ii}	LB	CRF	CRR	CCF	CCR	DRF	DRR	DCF	DCR
12	2	5	258	360	345	296	*283*	346	296	304	297
7	2	5	258	347	285	274	272	346	*270*	280	269
12	2	7	309	381	325	315	327	384	384	*310*	335
7	2	7	309	*310*	319	318	328	358	341	333	338
12	2	12	529	*529*	*529*	*529*	*529*	534	530	534	532

Table 3: Algorithm results for Case 2 requirements in Table 1